INTERNATIONAL JOURNAL OF eBUSINESS AND eGOVERNMENT STUDIES
Vol 10, No 1, 2018 ISSN: 2146-0744 (Online)

A CONCEPTUAL MODEL FOR AN SSM APPROACH TOWARDS THE IMPROVEMENT OF THE INSTRUCTIONAL DESIGN OF A COMPUTER SCIENCE MODULE

Suné van der Linde
North-West University, South Africa
sune.vanderlinde@nwu.ac.za

Malie Zeeman
North-West University, South Africa
malie.zeeman@nwu.ac.za

─Abstract ─

This article puts the topic of learning how to program in the spotlight. It remains difficult for students to learn how to program due to various challenges students experience which includes social and technical difficulties. The aim of this study is to develop a conceptual model for the improvement of the instructional design of a programming module in a computer science program using Soft Systems Methodology (SSM). This study applies the SSM cycle which will enquire the current instructional design in the programming module, formulate a relevant activity model showcasing a specific worldview (from a lecturer's perspective), using the model to further enquire the real-world situation and finally find changes that are both desirable and feasible based on the deeper understanding of the perceived problematic situation, i.e. learning how to program.

Key Words: SSM, Programming, Instructional design

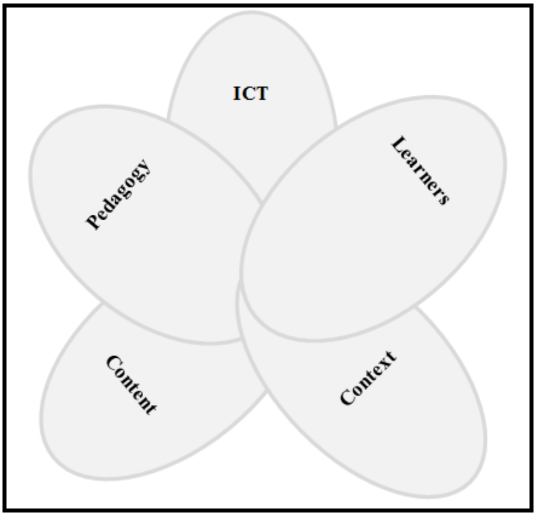
JEL Classification: I21, I23

1. INTRODUCTION

1.1 Learning to program

Learning to program is a challenge for students (Govender et al., 2014:187; Matthews et al., 2012:293; Robins et al., 2003:137). There are various reasons why students struggle with programming whether it is poorly constructed mental models (Ma et al., 2008:346), lack of problem solving skills (Govender et al., 2014:187; Havenga et al., 2013:5; Saeli et al., 2011:80), limited or no prior programming experience (Govender, 2010:14), inefficient learning styles (Raadt & Simon, 2011:111) or students' lack of belief in their own programming ability (self-efficacy) (Govender et al., 2014:189; Kinnunen & Simon, 2012:12). Many of these issues are not related to module content but rather to social and technical difficulties students experience while learning how to program. Teaching students how to program poses its own challenges. A model referred to as the technological pedagogical content knowledge (TPCK) model (figure 1) illustrates the complex environment and the different aspects such as ICT, content of the module, context and pedagogy as well learner specific issues that are involved in teaching programming and logical thinking skills (Ioannou & Angeli, 2016:2).

Figure-1: Technological Pedagogical Content Knowledge framework



Source: Ioannou and Angeli: 2016:2.

An improved instructional design considering all the factors involved in teaching programming skills could result in improved outcomes in terms of computer programming skills.

1.2 Motivation for SSM

The soft systems methodology (SSM) approach, which is based on social reality and human situations, aims to enquire a problematical situation, plan an intervention and take action in order to improve the situation (Checkland & Poulter, 2006:4; 2010 ; Checkland & Winter, 2006:1435). Since the SSM is based on social theory it is imperative to acknowledge that people have different worldviews or assumptions about the world. A person's view of the world is partly based on genetic inheritance and previous experiences but may change over time (Checkland & Poulter, 2006:4). The fact that people involved in the situation may see and experience things differently is fundamental in an attempt to improve a problematical situation (Checkland & Poulter, 2010:192). Many aspects which includes technical and social aspects must be taken into account when teaching students how to program. Therefore, teaching students how to program can be seen as a complex or "messy" situation. SSM allows the people involved in a problem situation to form conceptual models of the complicated "messy" environment they function in. Since programming education can be regarded as a "messy" and complex process, SSM is useful to articulate different viewpoints on the desired cause of action to improve the situation.

2. RESEARCH METHODOLOGY

The research was done within the critical research paradigm. Critical research is an extension of interpretivism in the sense that people are studied in relation to their world with the purpose of improving their lives or circumstances (Kuechler & Vaishnavi, 2011: 311; Neuman, 2011: 108). This study applies the SSM cycle (Checkland & Poulter, 2006) which enquires the instructional design currently being applied in teaching a programming module and formulate an activity model showcasing one prevailing worldview (a lecturer's perspective). The activity model is used to further enquire the programming education situation and finally identify changes to be affected that are both desirable and feasible based on the deeper understanding of the perceived problematical situation.

The SSM cycle entails five action steps which include enquiring about the perceived problematical situation (finding out step), constructing purposeful activity models, using the models to discuss the situation and possible improvements, and defining the action to be taken to bring about improvement.

3. APPLICATION AND RESULTS

This section pertains to the application of the five steps of the SSM used for the empirical part of this study. The results include a conceptual activity model for the improvement of the instructional design of an IT module.

3.1 Finding out

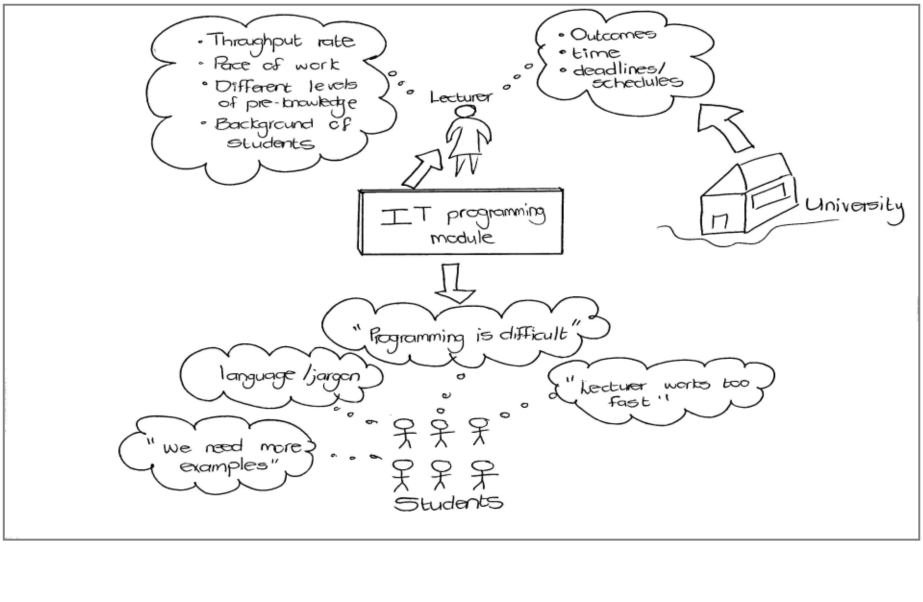
Checkland and Poulter (2006:24) suggest four techniques to perform the Finding out step. These are Making rich pictures, performing Analysis One (focus on intervention), performing Analysis Two (social analysis) and performing Analysis Three (political analysis). Each of these techniques is discussed next.

3.1.1 Making rich pictures

A picture shows relationships much better than a written paragraph (Checkland & Poulter, 2006:25). Figure 2 depicts the problematical situation and reflects the views of lecturers and students. Six lecturers with many years of experience in teaching computer programming skills participated in interviews as well as six computer science students who find programming a difficult skill to master. The leading questions during the interviews with lecturers were: "*Why do you think students have difficulty learning how to program?*" and "*How can these issues be addressed?*" Question posed to students during their interviews were: "*What are the challenges you face when learning how to program?*" and "*How can lectures in programming be improved towards positive outcomes?*"

During the interviews, various factors that play a role in the teaching and learning of a programming module were mentioned. The university has targets and time frames that lecturers need to adhere to. Therefore, students find it difficult to keep up. Some lecturers mentioned that they face challenges in terms of throughput rates (student performance) due to the lack of, and different levels of students' prior knowledge of technology. Students mentioned during the interviews that

Figure-2: Rich picture for the problematical situation pertaining this study



the pace of work is sometimes too fast. They also indicated that language barriers contribute towards poor performance in programming modules. Some students stated that they are not familiar with computer related terms (jargon) used in class. Others mentioned that more examples could improve their understanding of programming concepts.

These responses were used to enrich the rich picture in figure 2. The rich picture can be improved, and will become richer as more views and responses are added over time since the process will never be completed (Checkland & Poulter, 2006:25).

3.1.2 Analysis One

Analysis One, is concerned with intervention and focuses on three key roles present in any intervention, namely, the role of the client, the practitioner and the concerned or affected. It is important to note that the roles, norms and values may change over time (Checkland & Poulter, 2006:35).

The role descriptors are as follows:

- *Client* - Someone who causes the intervention
- *Practitioner* - Someone conducting the investigation
- *Affected* - The practitioner could list the concerned or people who are affected by the situation and outcome.

The roles assigned to the people involved in the intervention described in this study are provided in table 1. Feedback from students over a period of time causes the lecturer to become concerned and make an intervention, even if the intervention is only conceptually. In the problematical situation related to this study we identified the university as part of the concerned and/or affected due to the fact that the results of students influence the university directly in terms of the throughput rate.

Table 1: Identifying the roles in Analysis One for this study

Roles	**Assigned to**
Client	Lecturer, Student
Practitioner	Lecturer
Concerned or affected people	Students, Lecturer, University

3.1.3 Analysis Two

This analysis is concerned with the social texture of the problematical situation. There are three elements that assist in creating the social texture namely, roles, norms and values (Checkland & Poulter, 2006:33).

Roles in the Analysis Two phase refer to the difference in social positions of members of the group. Roles can be either formal or informal. A formal role can be, for example, head of department. Informal roles are formed in the social setting, for example, being known as a "trouble maker". In this study, we recognised only the formal roles of the lecturer, student and the University

Norms describe expected behaviours that are associated with a role. In table 2, the norms for each role are described.

Table 2: Norms associated with specific roles identified in Analysis Two

Roles	**Norms**
Lecturer	Professional, well prepared
Students	Attend class, prepare for class, complete assignments
University	Provide a safe environment, keep labs in order, provide tools to support learning such as an LMS

Values are the criteria by which the behaviour of roles is judged. We listed some of the values related to the roles listed in table 3 although there could be more values to list.

Table 3: Values for specific roles identified in Analysis Two

Roles	**Norms**	**Values**
Lecturer	Professional, well prepared	Be on time for class Be the expert on subject matter
Students	Attend class, prepare for class, complete assignments	Hand assignments in on time Participate in class
University	Provide a safe environment, provide a quality study environment, provide tools such as LMS	Keep labs in working order Appoint well qualified and competent lecturers.

3.1.4 Analysis Three

This analysis is concerned with the politics of the problematical situation, thus, the disposition of power (Checkland & Poulter, 2006:34). Here, a series of questions are asked about power where the term ‘commodity’ is used as a metaphor for power:

- What are the commodities that signal that power is possessed in a situation?
- What are the processes by which these commodities are obtained, used, protected, defended, etc.?

Having access to, or withholding important information can also be a commodity.

Based on Analysis Two, Analysis Three for this study is discussed next:

Lecturer – In the problematical situation at hand, the lecturer holds a certain level of power over students, even if it is only by the nature of the roles where the lecturer is naturally in a leading role in lecturing the content of a module and taking decisions required. For example, the lecturer has the power to decide which textbook to use, writes/modifies the study guides, decide on the method of delivery, the content to be included, etc. Students rely on lecturers for guidance on which resources to use. The lecturer has to ensure that the module outcomes are met on completion of the module. Furthermore, the lecturer has to make decisions that could affect students’ performance; for example, the lecturer creates a semester schedule with sections of the work to be completed each week. If one week was allocated (3 hours contact time) for introducing the *decision making* instruction (e.g. if statements), but some students are not on the level of understanding and applying the concept during the time allocated to this concept, the lecturer has to make a decision to either continue with the next section of work or spend more time on the decision making concept. No matter what decision is made, it will have an effect on the students as there will be less time to complete other sections of work that may need more time to complete. This becomes a problem at the end of the semester with all work might not yet being done or students not able to apply difficult concepts; hence the module outcomes might not be reached. The lecturer also has to consider throughput rates, which adds additional complexity and stress upon decisions to be taken.

University – the university possess power due to the nature of the situation. The university employs the lecturer, and the university can make decisions that affect lecturers. A vast number of possible dispositions of power exists from the perspective of the university, including the duration of contact time with students, enforcing alignment with other campuses which changes module outcomes, throughput rate targets, appointment of qualified lecturers to teach programming skills, etc.

Student – the students have the ability to choose the level of effort they put into each module, which will have a direct influence on their performance in a module. They can decide on whether to apply a positive or negative attitude towards the university, lecturer and the task at hand – learning how to program which could an effect on the lecturer and the university.

The following commodities are highlighted based on the problematical situation for this study:

- The lecturer makes decisions in terms of module content, delivery method, time schedule decisions, and textbook choices.
- The university dictates contact time, test series dates (which takes away contact time), module alignment with other campuses, the maintenance of the computer labs, the learning management system to be used etc.
- The student's attitude towards his/her studies plays a vital role in the outcome.

3.2 Making purposeful activity models

Purposeful activity models are used to create an organized process of enquiry (Checkland & Poulter, 2006:38). Such a model represents a single worldview and therefore it can never be considered as a real-world model (Checkland & Poulter, 2006:38). There are five steps involved (indicated in figure 4) in the construction of purposeful activity models. The subsequent section consists of a discussion on these five steps and the application thereof for the purpose of this study.

3.2.1 The PQR formula

A PQR formula can be used to write a root definition (RD) for a problem. The PQR formula entails the following:

P – what does the system do?

Q – how?

R – why?

For this study we identified PQR as follows:

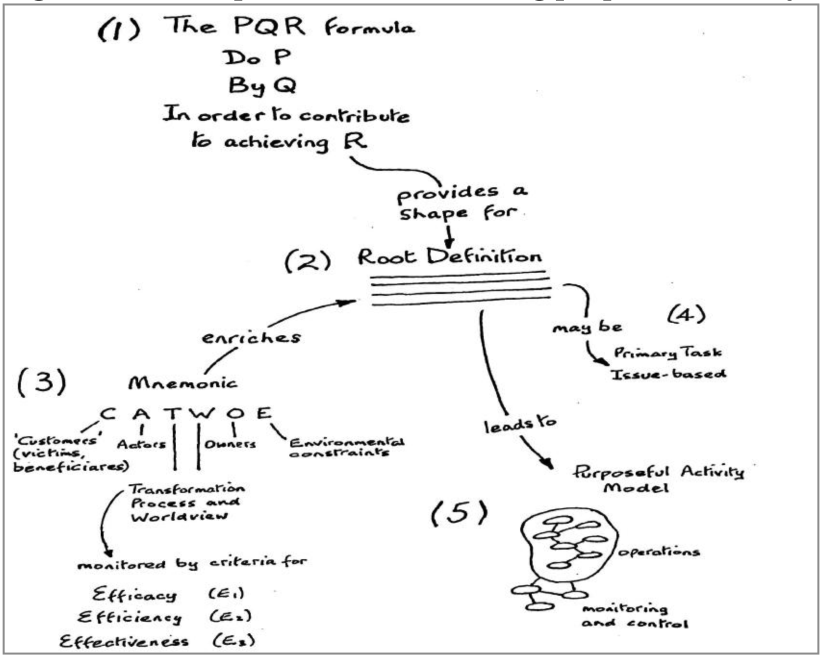
P – teach students the fundamentals of programming;

Q – instructional design;

R – in order to improve students' understanding, thus improving the module performance.

Checkland and Poulter (2006:39) suggest that the process of identification of PQR will assist greatly in the writing of the root definition (RD). The transformation process is captured in Q, thus, the instructional design. After reflecting on "instructional design" as the entire transformation process, it was clear that it is not descriptive enough and we had to revise the PQR.

Figure-4: Five steps towards constructing purposeful activity models



Source: Checkland and Poulter, 2006:38.

We decided that Q and R and should rather change to:

Vol 10, No 1, 2018 ISSN: 2146-0744 (Online)

P – teach students the fundamentals of programming;

Q – by changing the instructional design;

R – in order to improve students' understanding within the given time frame, thus improving the module performance.

3.2.2 The Root Definition

The root definition (RD) for this study based on the PQR formula is teaching students the fundamentals of programming within the given time frame by changing the instructional design in order to improve students' understanding, thus improving the module performance. Since the specific perspective (one way of looking at the situation) was not clear from the suggested RD we changed the RD as follows:

"*teaching students the fundamentals of programming within the given time frame by changing the instructional design in order to improve students' understanding, thus improving the module performance, from a lecturer's perspective*".

The RD affects the purposeful activity which needs to be modelled as a transformation process where instructional design for teaching programming skills should be transformed into a different state of the instructional design to encourage improved outcomes (Checkland & Poulter, 2006:41).

3.2.3 The mnemonic CATWOE

The mnemonic CATWOE is applied to describe exactly what is meant by the purposeful activity and is used to assist in the model building phase (Checkland & Poulter, 2006:41). It contains the following elements that should be considered when planning a purposeful activity:

- *C* – affected persons outside the system who will also benefit;
- *A* – people who perform the activities, which will make T, happen;
- *T* – transformation process;
- *W* – worldview;
- *O* – can be stopped by this person/s;
- *E* – constraints from the environment.

In this study, the elements to be considered when constructing the purposeful activity that could bring about change can be described as follows:

- *C* – students, university, parents, funding schemes;
- *A* – lecturer, students;
- *T* – changing the instructional design;
- *W* – a lecturer's perspective;
- *O* – head of department, dean, university;
- *E* – time constraints.

The experienced computer programming lecturers who participated in the study assisted in identifying changing the instructional design as the T element of CATWOE and provided us with different perspectives from a lecturer's point of view (W). They also identified time constraints (E) as one of the major issues in the programming education environment.

In any system, measures of performance are necessary to track the development and performance. The three E's as they are known give guidance to how a system will be judged (Checkland & Poulter, 2006:41):

- *Efficacy* – criteria to measure whether T is working;
- *Efficiency* – criteria to measure whether T is achieved with minimum resources;
- *Effectiveness* – criteria to measure whether T is helping to achieve a long-term aim.

This criteria needs to be set up, in order to measure the systems performance. For this study they are as follows:

- *Efficacy* – are student performance increasing?
- *Efficiency* – are outcomes reached within the required time frame?
- *Effectiveness* – are students' understanding of fundamental concepts improving?

3.2.4 Is the Root Definition a primary task or issue based?

Issue based models cut across organizational boundaries. These types of models encourage broader views and discussions (Checkland & Poulter, 2006:44). A primary task entails more isolated tasks. For this study the RD cuts across organizational boundaries. The nature of the problem investigated in this study does not only involve one aspect, for example, not only the module, but also the students and the university and therefor the root definition for this study was identified as being issue based.

3.2.5 Model building

Model building can be described as "putting together the activities needed to describe the transformation process" (Checkland & Poulter, 2006:44). Taking the first four steps of purposeful activity model building (3.2.1 to 3.2.4) into consideration and based on interviews conducted with the six experienced computer programming lecturers, and six students who participated in the study, our purposeful activity model from a lecturer's perspective has been constructed as shown in figure 4.

Our aim is to teach students programming fundamentals (P) in order to improve their programming skills within the given time frame, thus improving the module performance (R). We propose in this model to reach our goal by transforming our instructional design (Q) from a lecturer centred approach to a student centred approach. Useful ideas were exchanged during modelling of activities which lead to the introduction of a student centred approach towards teaching programming skills with specific reference to steps 3 and 4 in figure 5 in terms of taking transformative action towards improving programming skills.

In figure 5, the purposeful activity model describes the entire process of changing the instructional design of a computer science module by requiring of students to become familiar with a specific section the work before class and complete a test on the fundamental programming concepts of the particular study unit each week.

This process ensures that students come to class well prepared and that they will be able to identify work that they do not understand. The lecturer will have access to the results of the tests, which will assist in identifying and addressing possible misconceptions as soon as they arise. These activities (1-7) are all based on achieving the purpose, being to transform the module in order to improve performance. Step 7, 8 and 9 monitors the first 6 steps against defined measures of performance as discussed in section 3.2.3 (the three E's). The adaptive control can then make changes as necessary.

The model can change over time and is iterative in nature because the planned action might not yield the desired results.

3.3 Using models to structure discussion about the situation and improvement

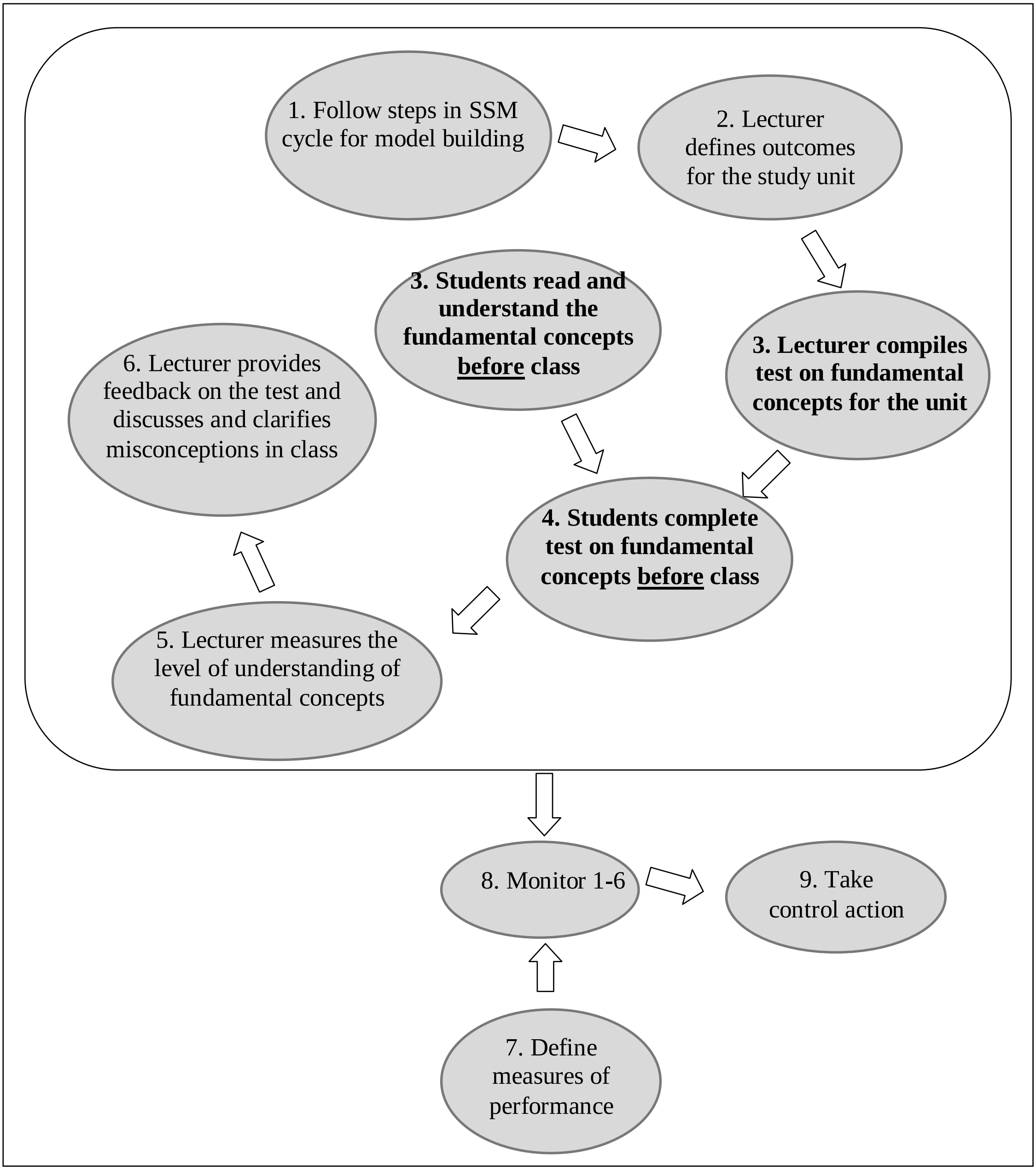
Checkland and Poulter (2006:51) suggest that discussions about the situation take place with the purposeful activity model present as a starting point for questioning. Thus we discussed the problem situation with the six experienced computer programming lecturers, who agreed that this model (figure 4) is a good starting point for progressing towards a learner centred approach towards teaching programming skills. During the discussion, we suggested and agreed that iteration of this process will cause change over time. New ideas or problems may arise from the implementation of the suggested model. Therefore it is recommended that more models should be drawn to incorporate new ideas or address the problems that may occur.

3.4 Defining action to improve

Checkland and Poulter (2006:55) points out that the aim of the previous step is not to find consensus amongst everyone, but rather finding accommodation. In order to physically transform a programming module, more rich pictures and models of purposeful activities needs to be drawn, so that other worldviews can also be taken into consideration.

We can only then seek accommodation after all other views have been taken into consideration. When that point is reached, action can be defined in order to improve the situation.

Figure-5: Purposeful activity model for changing the instructional design of a CS module



4. CONCLUSION

In this study we acknowledge the fact that programming is a difficult skill to master. Since many factors are involved programming education soft systems methodology was identified as a way to enquire about the problematical situation, plan and apply an intervention. The enquiry phase was most informative since different lecturers have different views and experience in the field of programming education that contributed towards understanding and articulating the aspects that contribute towards students not performing well in programming modules. The interviews with students also gave another perspective and should be explored in more detail in future research. Useful ideas were exchanged during modelling of activities that lead to the introduction of a student centred approach towards teaching programming skills.

In terms of the application of the SSM we found that the graphical representation of the situation and the holistic approach towards the problem contributes towards a better understanding of the problematical situation from a social and technical point of view. Using the SSM guide the user towards consciously thinking about the problem and to gain more insight than simply recognising the problem that exists.

The SSM can be applied to investigate and intervene to improve any problematical situation in a social context. In this study a conceptual model was designed that aims to improve the outcomes when teaching programming skills. Since low throughput figures due to poor understanding of programming skills is a matter of concern we aim to implement the model as an intervention to improve the situation.

REFERENCES

Checkland, P. & Poulter, J. 2006. *Learning for action: a short definitive account of soft systems methodology and its use for practitioner, teachers, and students,* Wiley Chichester.

Checkland, P. & Poulter, J. 2010. Soft systems methodology. *Systems approaches to managing change: A practical guide.* Springer.

Checkland, P. & Winter, M. 2006. Process and content: two ways of using SSM. *Journal of the Operational Research Society,* 57**,** 1435-1441.

Govender, I. 2010. From Procedural to Object-Oriented Programming (OOP)- Performance in OOP: An empirical study. *South African Computer Journal,* 46.

Govender, I., Govender, D. W., Havenga, M., Mentz, E., Breed, B., Dignum, F. & Dignum, V. 2014. Increasing self-efficacy in learning to program: exploring the benefits of explicit instruction for problem solving. *TD: The Journal for Transdisciplinary Research in Southern Africa,* 10**,** 187-200.

Havenga, M., Breed, B. & Mentz, E. 2013. Metacognitive and Problem-Solving Skills to Promote Self-Directed Learning in Computer Programming: Teachers' Experiences. *SA-eDUC,* 10.

Ioannou, I. & Angeli, C. A Framework and an Instructional Design Model for the Development of Students' Computational and Algorithmic Thinking. MCIS, 2016. 19.

Kinnunen, P. & Simon, B. 2012. My program is ok – am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education,* 22**,** 1-28.

Kuechler, B. & Vaishnavi, V. 2011. Promoting relevance in IS research: An informing system for design science research. *Informing Science: The International Journal of an Emerging Transdiscipline,* 14**,** 125-138.

Ma, L., Ferguson, J. D., Roper, M., Ross, I. & Wood, M. Using cognitive conflict and visualisation to improve mental models held by novice programmers. ACM SIGCSE Bulletin, 2008. ACM, 342-346.

Matthews, R., Hin, H. S. & Choo, K. A. 2012. Merits and pitfalls of programming learning objects: a pilot study. *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia.* Bali, Indonesia: ACM.

Neuman, W. L. 2011. *Social research methods – qualitative and quantitative approaches.,* Boston, MA:Pearson Education.

Raadt, M. D. & Simon 2011. My students don't learn the way I do. *Proceedings of the Thirteenth Australasian Computing Education Conference - Volume 114.* Perth, Australia: Australian Computer Society, Inc.

Robins, A., Rountree, J. & Rountree, N. 2003. Learning and teaching programming: A review and discussion. *Computer Science Education,* 13**,** 137-172.

Saeli, M., Perrenet, J., Jochems, W. M. & Zwaneveld, B. 2011. Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education,* 10**,** 73-88.